Authentication using JWT – Json Web Token

Step 1 :

```
app.post("/login22", async function(req, res){

        const user = await Users.checkCrediantialsDb(req.body.username,
req.body.password)
        const token = await  user.generateAuthToken()

})
```

Step 2 :

Create a function in the user model validate the user

```
userSchema.statics.checkCrediantialsDb = async (user22, pass) =>{

 const user1 = await User.findOne({name : user22, age : pass})

   return user1;

}
```

Step 3 :

Create a function that generates the auth token

```
userSchema.methods.generateAuthToken = async function () {
    const user = this
const token = jwt.sign({ _id: user._id.toString() }, 'thisismynewcourse')

console.log(token);
    user.tokens = user.tokens.concat({ token :token })
    await user.save()

    return token
}
```

Step 4 :

Create a file called auth.js in middleware folder

```
const jwt = require('jsonwebtoken')
const User = require('../models/user')

const auth = async (req, res, next) => {
    try {
        const token = req.header('Authorization').replace('Bearer ', '')
        const decoded = jwt.verify(token, 'thisismynewcourse')
        const user = await User.findOne({ _id: decoded._id, 'tokens.token': token
})

        if (!user) {
            throw new Error()
        }

        req.token = token
        req.user = user
        next()
    } catch (e) {
        res.status(401).send({ error: 'Please authenticate.' })
    }
}

module.exports = auth
```

Step 5 :

Require the auth.js in main js file

```
const auth = require('./middleware/auth');
```

Step 6 :

Now you can control the route using the auth middleware as follows

```
app.get("/test99", auth, function(req, res){

    })
```